Udacity Data Engineering Nanodegree Capstone Project

John Maxi

**Step 1: Scope the Project and Gather Data**

The goal of my project is to build an analytics table for researching the relationship between air quality and person-level health outcomes. Air quality data will be obtained from the EPA <https://aqs.epa.gov/aqsweb/airdata/download_files.html> and person-level health outcome will be obtained from the Nation Health Interview Survey <https://www.cdc.gov/nchs/nhis/data-questionnaires-documentation.htm>. I have collected data from 2014 onwards for the EPA data, and 2015 onwards for the NHIS data. In total, the raw data contains 10s of millions of rows.

The goal of the data model is to be able to explore spatial-temporal relationships between the air quality and the health outcomes. The data model will need to have the capacity to marry the time and location of the air quality measurements with the time and location of the health outcomes data.

**Step 2: Explore and Assess the Data**

The data exploration can be viewed in detail in the file: Step 2 - EDA and Data Cleaning. To clean the data I took the following steps for each dataset:

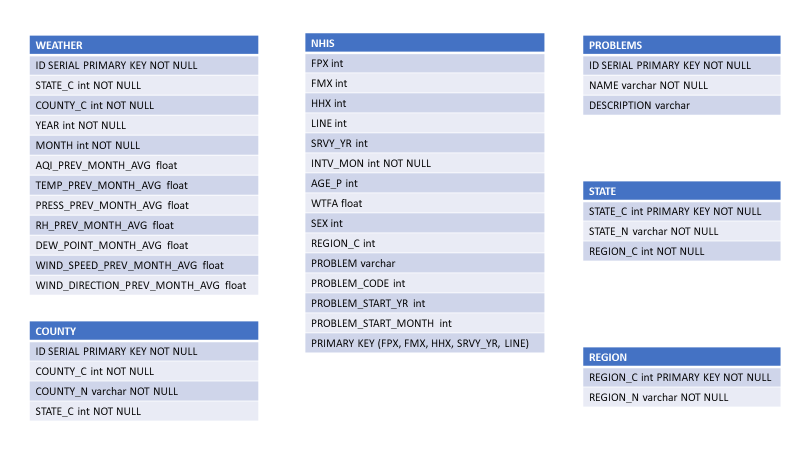
Weather data:

1. Open and concatenate each type of weather variable file into a single table.
2. Filter the PM 2.5 rows so as to keep only the rows that are measured in the accepted standard.
3. Convert date strings to datetimes for processing.
4. Extract the unique state code and names in the dataset and map the states to their respective regions based on: <https://www.businessinsider.com/regions-of-united-states-2018-5>. The only geographic information available in the NHIS dataset is the region of the country. Creating the map from state to region is necessary to overlap the locations of the sensors and health outcomes. The granularity of the NHIS data necessitates that the final analysis will cover broad swaths of the country. States that are mapped to one of the four regions, e.g., Puerto Rico, their region value is set to -1.
5. The monthly average values at each location are calculated for each of the weather variables (air quality, humidity, dew point, temperature, wind speed, wind direction, pressure).
6. Next, a date offset of one month is added to the datetime value associated with the average weather variables. This causes the weather data to be one month in the past relative to its corresponding time stamp. This is performed because we want to be able to investigate how the past months weather affected health problems in the next month.
7. Finally, I have chosen to store the year and month associated with the readings as two columns representing the year and the month respectively, rather than using a datetime value. This is because the granularity of the NHIS data is only year and month. Storing more detailed time data would be a false representation of the data. (Similar to choosing the correct number of significant digits in science. We wouldn’t report the number of digits the calculator is capable of displaying, we would only display as many digits as our instruments are capable of measuring).

NHIS data:

1. Drop data rows that have been flagged as poor quality by NHIS.
2. Each of the different types of health problems recorded by the survey are given their own column in the raw data, along with columns describing how the long the problems have lasted. Additionally, problems for children are separate from problems for adults. For each combination of problem type, duration, and survey date, I calculate when the problem started relative to the month the survey occurred for that person.
3. A single person in the NHIS survey is represented by four variables: person number, family number, house hold number and survey year. Since I am processing the data in yearly data files, I use the first three variables to determine how many times a given entity occurs in the data. These values become the LINE value in the final data model.
4. Next, I extract the year and month the problem started as integer values like in the weather table.
5. In the raw data, all problems associated with children are distinct from problems associated with adults, even if the problems are similar (e.g., there is a vision problem code for adults, and separate one for children). I combine the similar problems into a single code. The end user will be left to filter on the age column if they want to separate out children from adults for these problems.
6. Finally, the problem codes are extracted from the raw data and mapped to their description. This becomes the problem dimension table in the final data model.

**Step 3: Define the Data Model**

Figure 1: ER diagram of the data model.

The chosen data model allows for combining information from different sources based on time and location. There are two fact tables, WEATHER and NHIS. WEATHER holds the information from air quality sensors summarized by month and location. The values associated with each row (a time and location) indicate the average value for each measurement type from the previous month at that location. The year and month columns indicate the current month. In turn, year and month columns can be used to either the year and month a person was surveyed using the SRVY\_YR and INTV\_MON in the NHIS fact table, or the time a person identified a problem as starting using the PROBLEM\_START\_YR and PROBLEM\_START\_MONTH columns. NHIS holds the responses of participants in the NHIS study regarding their medical problems. Each row represents a person’s response to a given type of health problem. NHIS includes the variable REGION\_C that can be used to connect the location of a person in the survey to the location of a sensor. Note that the sensor information must be rolled up to the region level using the appropriate dimension tables, COUNTY, STATE and/or REGION.

Pipeline steps:

1. Download data from sources:
   1. Weather source
   2. Nhis source

**Step 4: Run ETL to Model the** Data

Write out the data dictionary

Add logging to the processing scripts and include the key outputs

* Primary keys are defined in the insert statements, along with what to do on conflict. Explain here
* Key logging outputs: number of rows per file, time take per processing function call
* Write unit test cases for each helper function
* Write function to get the count/shape of each finished table to be sure it matches what went into the insert statements (no data got dropped during insert)

Air quality data: The primary key will be the daily timestamp, the columns will indicate the different measured values for each day: PM2.5, pressure, temp, humidity, wind. There are also going to be many locations per variable, so I will need to take that into account as well. Key will need to end up including both the time and space distribution of the readings.

Should I have 2 fact tables or just one? For an “analytics table” use case I should distill all the information into a single star schema. For a “source-of-truth” database I think I want to keep the raw data, so I would have the air data table with dimensions and the health data table with dimensions. Would I be joining them with the dimensions?

A monitor is indicated by the site (state + county + site number), pollutant code and POC

PM 2.5:

State Code: Int, range 1-80

County Code: Int, range 1-810

Site Num: Int, range 1-9997

Parameter Code: Int, same value for all rows in a yearly file

POC: Int, range 1-33. POC stands for parameter occurrence code. Used to uniquely identify a monitor if there is more than one device measuring the same pollutant at the same site.

Latitude: numeric

Longitude: numeric

Datum: varchar, NAD83 or WGS84. What is this? It is associated with the latitude and longitude

Parameter: varchar, same for all rows of a yearly PM2.5 file 2.5 – Local Conditions. I think this probably corresponds to the Parameter column. Probably each parameter matches the data type per file (so humidity has a different parameter number).

Sample Duration: varchar, 1 HOUR, 24 HOUR, 24-HR BLK AVG. This refers to the length of time air flows though a monitor before being analyzed. It represents an averaging period.

Pollutant Standard: varchar, PM25 24-hour 2012 or blank. This indicates how the pollutant values are calculated. The EPA standard is 24 hour averages. Some of the rows are 1 hour duration averages

Date Local: date

Units of Measure: varchar

Event type: Indicates if data is included after an uncontrollable event occurred (e.g., wildfire)

Observation Count: int, number of observations taken that day

Observation Percent: int, number of observations taken with respect to the number of scheduled for that day

Arithmetic mean: numeric, the average value for the day

1st Max Value: numeric, the highest value for the day. Only meaningful when the sample duration is 1 hour

1st Max Hour: int, the hour at which the 1st Max Value occurred. Is always 0 in the daily summery. Only meaningful when the sample duration is 1 hour

AQI: Air Quality Index for the day

Method code: int, internal code for identifying the processes/protocol of data collection

Method name: varchar

Local Site Name: varchar, name of the site given by local operators

Address: varchar

State Name: varchar

County: varchar

City Name: varchar

CBSA Name: varchar, core bases statistical area (metropolitan area where the monitoring site is located

Date of Last Change: date, the last time any numeric value was changed in the record

I’m thinking of a data model that splits this table into a fact table – readings, and dimension tables for time and place aspects of the data. At this point I’m planning to only keep the 24 data so I won’t need the 1st max or 1st max hour columns at all.